PLC Security
TOP 20 LIST

工控PLC安全编码最佳实践

(中文版)

NSFOCUS

格物实验室
NSFOCUS GEWU LAB

译者序

从 2010 年的“震网”事件后，工控安全走进了大众视野，第一次让大家认识到远离我们日常生活的关键基础设施遭受网络攻击的后果，如果没有将关键基础设施纳入到网络安全防护的体系中，我们的国计民生将会受到严重威胁，轻则停水停电，重则发生爆炸扰乱社会正常秩序。因此，我国也在 2016 年发布了《工业控制系统信息安全防护指南》，自此之后也涌现出了一批工控安全相关的产品，用以防护控制关键基础设施的重点工业控制系统。

工控安全防护体系的路线经历了最初的“隔离即安全”，通过构建无法访问互联网的封闭网络来保护重要资产；接下来是“被动式的单点防护”，当边界概念愈发模糊、两化融合更加深入时，必须要采取安全措施，此时企业限于预算/意识等原因，采取被动式的、单一的边界/主机防护、检测、监测类产品来保护重要资产；第三阶段是“多层次、多维度的纵深防御”，这个阶段中需要从管理、技术、运维多个维度出发，结合 IT 领域的经验，部署对应的产品、平台等，形成综合防御、立体监控，以期达到保护工控业务和资产；第四阶段是“内生安全”，从控制系统的内部出发，建立工控系统全生命周期安全管理，融合安全开发流程和可信计算、国密等新技术，以消除系统各个生命阶段可能出现的安全风险；第五阶段是“以攻为守的技术震慑”，此阶段类似于大家都拥有核武器，以威慑对手不敢轻而易举地攻击我们。这个阶段需要提高工控领域的攻防水平，形成技术震慑力，同时储备战略资源，建立了多个工业控制系统安全研究实验室，以高度仿真的工控系统攻防平台为基础，通过对典型工控网络威胁的复现、工控系统脆弱性的研究，提高攻击技术，形成威慑力。虽然以上的防护路线描绘了每个阶段应该落实的重点工作，还是忽略了一点，我们都是从外部或者内部的视角去审视如何抵御攻击的，但当我们再次去理解工业控制系统时，发现其实它是一套编程人员开发出来的用以完成特定工业过程的设备或者系统，如果最开始由编程人员编写的程序就存在一些安全风险，即使我们后续加码各类防护手段，还是会留下一个明显的脆弱点。那么怎么才能避免这部分的风险呢？

凡是可编程的语言都有它的安全编码规范，比如 C、C++、java 等诸多计算机语言安全编码规范，其实工业控制系统也是一个可编程的系统，他们也应该拥有自己的安全编码规范或者代码风险管理措施，基于此，我们翻译了由 ISA 下属的网络安全小组发起的社区项目《Top_20_Secure_PLC_Coding_Practices_V1.0》，目的是让我们国内也意识到，控制系统的安全编码实践也是构建工业控制系统安全防护体系必不可少的一个环节。

目前工业控制系统的逻辑程序大部分由承包项目的集成商或者工控厂商的工程人员完成，这部分工程师水平参差不齐，而且项目验收中也只是关注工业控制系统的功能是否正常，没有针对内部的逻辑程序做审查，因此也存在一定的风险，比如程序中使用了数组，数组的下标是一个变量，但是没有对这个变量做溢出检查。所以很有必要针对控制系统的编码过程做出要求，用来消除不必要的错误或风险。

我们将工业控制系统编码规范化、安全化后有什么效果呢？是不是就可以阻止网络攻击呢？答案是否定的，安全编码规范并不能完全阻止网络攻击，但是它可以避免工业控制系统运行过程中出现的各类错误，为调查相关安全事件提供一些参考。阻止网络攻击并不是某一个环节、一个设备就能完成的，它需要从工控系统本体、安全编码、外部防护设备、软件系统、管理运维、应急响应等多个体系或者维度一起努力才能达到的最终安全防护的终极目标。

——高 剑 田泽夏

2021 年 9 月

1.PLC 代码模块化

将 PLC 代码拆分成不同模块，例如不同的功能块（子程序等），每个模块进行独立的测试验证。

2. 跟踪并监视运行模式

确保 PLC 保持在 RUN 模式，一旦运行模式发生变化，应发出相应的告警。

3. 尽可能让 PLC 处理更多的逻辑

尽量将运行逻辑放在在 PLC 中，例如累加计数和乘积运算等。 因为 HMI 目前还无法很好地实时的处理类似的运算任务。

4. 利用 PLC 中的特殊标志来验证完整性

在 PLC 的各类错误标志中加入计数器，以此来捕获潜在的编码问题。

5. 利用加密方式或者校验和的方式来验证 PLC 代码的完整性

利用加密散列或者校验和来保证 PLC 代码的完整性，并且在代码发生变化时发出告警。

6. 验证定时器和计数器

如果定时器和计数器的值被写入 PLC 程序，PLC 应该验证写入值的合理性，并且验证输入值小于零时计数器反向计数情况。

7. 对成对的输入/输出进行验证和警报

如果有成对的信号，确保两个信号没有同时被使用或者激活。当出现物理上不可行的输入/输出状态时，应向操作员发出告警。当切换输出可能损害执行器时，应考虑独立处理对应的信号或添加延迟计时器以避免出现事故。

8. HMI 的输入变量应该同时在 HMI 和 PLC 层面进行验证。

HMI 对 PLC 变量的访问可以(也应该)限制在 HMI 的有效操作值范围内，但应在 PLC 中添加进一步的验证检查，以防止接收超出范围的值，如果出现应及时发出告警。

9. 间接验证

通过对数组的溢出或异常下标输入来验证间接指令，以捕捉“栅栏柱（fence-post）”错误。

10. 通过函数分配指定的寄存器块(读/写/验证)

为特定函数分配指定的寄存器块，以验证数据，避免缓冲区溢出和阻止未授权的外部写操作，以保护控制器数据。

11. 合理性检查工具

引用一种检查流程，利用交叉检查来验证输入数据或者逻辑的合理性。

12. 基于物理层面的合理性来验证输入

确保操作员在输入过程中只能输入实际有意义的数据。当出现偏差时或者无效输入时，应及时发出告警。

13. 禁用不需要/不用的通信端口和协议

PLC 控制器和网络接口模块一般支持默认开启的多种通信协议。应该禁用设备中不需要的端口和协议。

14. 限制第三方数据接口

限制第三方接口的连接类型和可用数据。应该对连接（和/或）数据接口进行适当定义，并将限制仅为所需数据进行读/写。

15. 定义 PLC 重新启动时的安全状态

在 PLC 重新启动的情况下，定义一个重启后的安全状态(例如，通电，断电，保持以前的状态)。

16. 收集 PLC 的运行周期并在人机界面上进行趋势分析

每 2-3 秒汇总 PLC 的执行周期时间，并上传给人机界面以图形化的方式呈现。

17. 记录 PLC 的正常运行时间并在人机界面上进行趋势分析

记录 PLC 的正常运行时间，以知道它重新启动的时间。在 HMI 上追踪和记录正常运行时间，用以进行诊断。

18. 在 HMI 上记录 PLC 硬停止并且进行趋势分析

在 HMI 上存储由故障或关机造成的 PLC 停止事件，PLC 重启前，由 HMI 发出告警。同时利用时间同步来获取更准确的数据。

19. 在 HMI 上监控 PLC 内存的使用情况并进行趋势分析

测量部署在生产环境中的每个控制器的内存使用情况并且建立一个基线，同时在 HMI 上对其进行趋势分析。

20.捕捉关键警报的假阴性和假阳性

识别关键告警并为这些告警设置门限 trap。通过门限 trap 来监控触发条件和有变化的告警状态。

1. PLC 代码模块化

将 **PLC** 代码拆分成不同模块，例如不同的功能块（子程序等），每个模块进行独立的测试验证。

安全目的	目标群体
PLC 逻辑的完整性	**产品供应商**

指导

不要在一个位置编写完整的 PLC 逻辑程序，例如在主组织块或主程序中。相反的，将其划分为不同的函数块(子程序)，并监控它们的执行时间和大小(以 Kb 为单位)。

为独立运行的逻辑创建单独的段。这有助于输入验证、访问控制管理、完整性验证等。

模块化的代码还有助于测试和跟踪代码模块的完整性。如果模块内部的代码已经经过精心测试，那么对这些模块的任何修改都可以根据原始代码的哈希值进行验证，例如，通过保存每个这些模块的哈希值(当这是 PLC 中的一个选项时)。通过这种方式，模块可以在 FAT/SAT 期间进行验证，或者在某些异常后，当代码的完整性存在问题时进行验证。

示例

燃气轮机逻辑分为“启动”、“进口导叶控制”、“排气阀控制”等，因此可以系统地应用标准代码块的逻辑。如果发生安全事故，这也有助于快速排除故障。

经过严格测试的自定义功能模块可以在不更改的情况下复用(如果尝试更改则会发出告警)，并通过密码/数字签名锁定防止滥用/误用。

优点与好处

维度	原因
安全性	**有助于检测可能是恶意代码的新增部分。有助于逻辑标准化、一致性和锁定未授权的修改。**
可靠性	**帮助控制程序流序列并避免可能导致逻辑不能正确执行或崩溃的循环。**
运维	**模块化代码不仅易于调试(模块可以独立测试)，而且易于维护和更新。此外，这些模块可用于其他 PLC，从而允许在单独的 PLC 中使用和识别公共代码。便于维护人员在故障处理过程中快速识别常用模块。**

参考材料

标准/框架名称	涉及内容
MITRE ATT&CK for ICS	Tactic: TA002 - Execution Technique: T0844 - Program Organization Units
ISA 62443-3-3	SR3.4: Software and information integrity
ISA 62443-4-2	CR3.4: Software and information integrity
ISA 62443-4-1	SI-2: Secure coding standards
MITRE CWE	CWE-1120: Excessive Code Complexity CWE-653: Insufficient Compartmentalization

2. 跟踪并监视运行模式

确保 **PLC** 保持在 **RUN** 模式，一旦运行模式发生变化，应发出相应的告警。

安全目的	目标群体
PLC 逻辑的完整性	**集成/维护服务提供商资产所有者**

指导

如果 PLC 没有处于 RUN 模式(例如，PROGRAM 模式)，那么可以通过重新编写代码来跟踪 RUN 模式。PLC 可以利用校验和或者其他方法在代码发生变化时发出告警，如果没有的话，至少要有一个间接的指标或者标志，可以用来跟踪操作模式中潜在的问题：

- 如果 PLC 未处于 RUN 模式，应向操作人员发出告警。如果目标控制系统上确实有维护人员在操作，维护人员应该确认告警并继续进行。
- HMI 应配置为在交接班时再次提醒操作员查看已存在的告警。目标应该是跟踪工厂中任何可能影响 PLC 操作模式的工作人员或承包商。

例外情况:如果电站处于测试或开发阶段，考虑禁用此告警，此时电站应与外部网络隔离。

示例

如果 PLC 没有改变工作模式的硬件开关，建议至少利用可以限制改变 PLC 代码的软件机制，例如设置 PLC 代码的密码保护机制。

优点与好处

维度	原因
安全性	**运行模式(运行/编辑/写入;Allen Bradley PLC: RUN / PROGram / REMote)影响 PLC 是否可以被篡改。如果按键开关处于 REMote 状态，即使 PLC 正在运行，也可以通过通信接口对 PLC 程序进行更改。**
可靠性	**/**
运维	**/**

参考材料

标准/框架名称	涉及内容
MITRE ATT&CK for ICS	Tactic: TA009 - Inhibit Response Function Technique: T0858 - Utilize/Change Operating Mode

3. 尽可能让 PLC 处理更多的逻辑

尽量将运行逻辑放在在 **PLC** 中，例如累加计数和乘积运算等。 因为 **HMI** 目前还无法很好地实时的处理类似的运算任务。

安全目的	目标群体
PLC 逻辑的完整性	**产品供应商、集成/维护服务提供商资产所有者**

指导

人机界面（HMI）提供了一定程度的编码功能，最初旨在帮助操作员增强可视化和告警功能，一些程序员使用这些功能创建的代码本应该放置在 PLC 中，以保持完整性和可审计性。

越接近实际现场的计算值越准确。HMI 无法及时更新 来保证求和 /乘积运算的有效性，而且 HMI 和 PLC 之间总是存在通讯延迟。此外，当代码放置在 PLC 中，HMI 重启时，它仍然可以从 PLC 接收求和和乘积运算值，而不至于重新开始计算。

特别要注意的是，应该避免任何与安全或安全功能(如联锁、定时、保持等)有关的 HMI 代码。

对于一段时间内的数据分析， 过程数据的历史记录比在 HMI 上记录更合适。在历史数据库中查询累计值(在一段时间内、在一个批处理中、在一个过程周期内)与在 PLC 逻辑中的汇总数值作比较，对差异较大的告警，一般都是由于数据采集粒度的差异造成的。

示例

应该有相应的代码来限制按钮的启用/禁用:启用/禁用操作应放置在 PLC 层，否则，这些操作不满足(预期的)条件时也可以在控制 PLC 的 HMI(或通过网络)上执行。

对操作人员的一些行动设置定时器(例如电机连续启动应有延迟定时器，阀门关闭/打开或电机停止的操作也应考虑加入定时器)不应该放在 HMI 层，而是应该在 PLC 层来控制这样的电机/阀门。

告警阈值可以显示在人机界面上，但这个阈值必须作为 PLC 代码的一部分用于处理非预期的输入值。

体积数值可变的水箱:由 PLC 控制水箱流量的进出，很容易计算出实际体积值(和交叉验证总数)。HMI 也能做到，但首先需要从 PLC 中获取相关数值。这些值需要准确的时间戳，从而在出现延迟或 HMI 重启时获取到正确的体积数。

优点与好处

维度	原因
安全性	**1.验证代码变更可以保持一致性。除了 PLC,HMI 编码也需要代码的变更控制，一般没有那么严格(特别是在施工阶段和调试阶段)，无法让客户拥有一个完整的视图，有可能还会忽略一些重要的考量条件。HMI 没有像 PLC 或 SCADA 那样的“强制信号”或更改值列表，因此 HMI 级别的更改更是难以检测，实际上不可能作为授权更改管理计划的一部分。** **2. 对于攻击者来说，攻击分布在多个 PLC 上的累计流量值要比操纵所有在 HMI 中计算的累计流量值更难。**

维度	原因
	3.如果部分启用/禁用功能不在 PLC 中，攻击者可能不需要 HMI 部分就可以操纵 PLC 和 I/O，因为这些信息在 HMI 中的话就已经被攻陷了。
可靠性	**1. 越接近实际现场，计算就会越加有效和准确。此外，如果 HMI 重启，总数和计数仍然有效可用(PLC 不经常重启，通常将这些值存储在非易失性存储器中)。** **2. 输入的来源不同可能意味着后续处理环节会有非预期的故障出现。工厂的人机界面可能有不同的产品形态和技术实现(SCADA 层，也包括现场触摸屏、控制器面板)，其中单一技术或者产品有可能无法将现场的信号变化将传输到其他层级，导致可视化的不一致性和潜在的操作故障。**
运维	**PLC 之间的编码更容易移植和互操作，HMI 之间的编码相对较难。**

参考材料

标准/框架名称	涉及内容
MITRE ATT&CK for ICS	Tactic: TA010 - Impair Process Control Technique: T0836 - Modify Parameter
ISA 62443-3-3	SR3.6 : Deterministic Output
ISA 62443-4-2	CR3.6 : Deterministic Output

4. 利用 PLC 中的特殊标志来验证完整性

在 **PLC** 的各类错误标志中加入计数器，以此来捕获潜在的编码问题。

安全目的	目标群体
PLC 逻辑的完整性	**产品供应商、集成/维护服务提供商资产所有者**

指导

如果 PLC 代码运行正常但突然做了除零操作，或者某个 HMI 正在从另一个 PLC 进行点对点通信，并且功能/逻辑在意料之外的时候做了一个除数为零的操作，这些异常行为都需要详细调查。

大多数程序员会忽略该问题，他们不认为自己的代码有问题。在代码开发期间，工程师需要通过输入超出预期范围的数据来测试和验证他们的代码模块(片段或例程)。这可以称为单元测试（unit level test）。

为固件、逻辑和协议栈分配不同的锁定内存段。测试协议栈的滥用情况，例如一个数据包报头中的特殊标志条件等。

示例

由数据越界引起的 PLC 故障是很常见的。例如，当输入值导致数组索引越界时，或带有反向功能的计时器时，或除零异常时，就会发生这种情况。

典型错误标志是:

- 除 0
- 计数器溢出
- 反向计数器或定时器预置
- I /O 扫描溢出

优点与好处

维度	原因
安全性	**对 PLC 的攻击可能包括改变其逻辑、激活新程序、测试新代码、加载新程序、插入辅助逻辑发送消息或激活某些功能。由于大多数 PLC 不提供加密、完整性检查，如果发生上述逻辑之一的更改，flags 可以是一个很好的指示。**
可靠性	**严谨认真对待 flags 可以避免 PLC 运行与编程或 I/O 错误。此外，如果发生错误，故障的来源就会更加明显。**
运维	

参考材料

标准/框架名称	涉及内容
MITRE ATT&CK for ICS	Tactic : TA010 - Impair Process Control Technique: T0836 - Modify Parameter
ISA 62443-3-3	SR3.5: Input Validation SR3.6: Deterministic Output
ISA 62443-4-2	CR3.5: Input Validation CR3.6: Deterministic Output
ISA 62443-4-1	SI-2: Secure coding standards SVV-1: Security requirements testing
MITRE CWE	CWE-128:Wrap-around CWE-190:Integer Overflow CWE-369:Divide by Zero CWE-754: Improper Check for Unusual or Exceptional Conditions

5. 利用加密方式或者校验和的方式来验证 PLC 代码的完整性

利用加密散列或者校验和来保证 **PLC** 代码的完整性，并且在代码发生变化时发出告警。

安全目的	目标群体
PLC 逻辑的完整性	**产品供应商、集成/维护服务提供商资产所有者**

指导

1. Checksum 校验和

在(加密)哈希不可行的情况下，可以选择校验和（checksum）。一些 PLC 在代码下载到 PLC 硬件时会产生一个唯一的校验和。校验和应在 SAT（现场验收测试）后由制造商/集成商记录，并作为保修/服务条件的一部分。

如果控制器本身没有校验和的功能，也可以在 EWS/HMI 中生成校验和并进行探测，例如，每天一次，与 PLC 中原始代码的哈希值进行比较，以验证它们是否一致。虽然这不会提供实时告警，但可以检测是否有人试图更改 PLC 代码。

校验和的值也可以放到 PLC 寄存器中，当它改变时，可以选择发出告警或者将异常值发送给历史数据库等。

2. Hashes 哈希散列

PLC cpu 在运行时一般没有生成或检查哈希的处理能力。尝试哈希实际上可能会导致 PLC 崩溃。但是 PLC 的组态软件可能能够从 PLC 代码中计算哈希值，并将它们保存在 PLC 或控制系统的其他地方。

示例

已知具有校验和特性的 PLC 供应商:

- 西门子(见示例)
- 罗克韦尔

此外，外部软件可用于生成校验和:

- Version dog
- Asset Guardian
- PAS

西门子的实现示例

在 Siemens S7-1500 PLC 中创建校验和的示例:

GetChecksum-Function Block 读取实际的校验和，通过一个轻量级脚本，“SATChecksum”可以作为引用存储。引用校验和的偏差（Reference-Checksum）可以存储在数据日志函数中（Datalog-Function）。

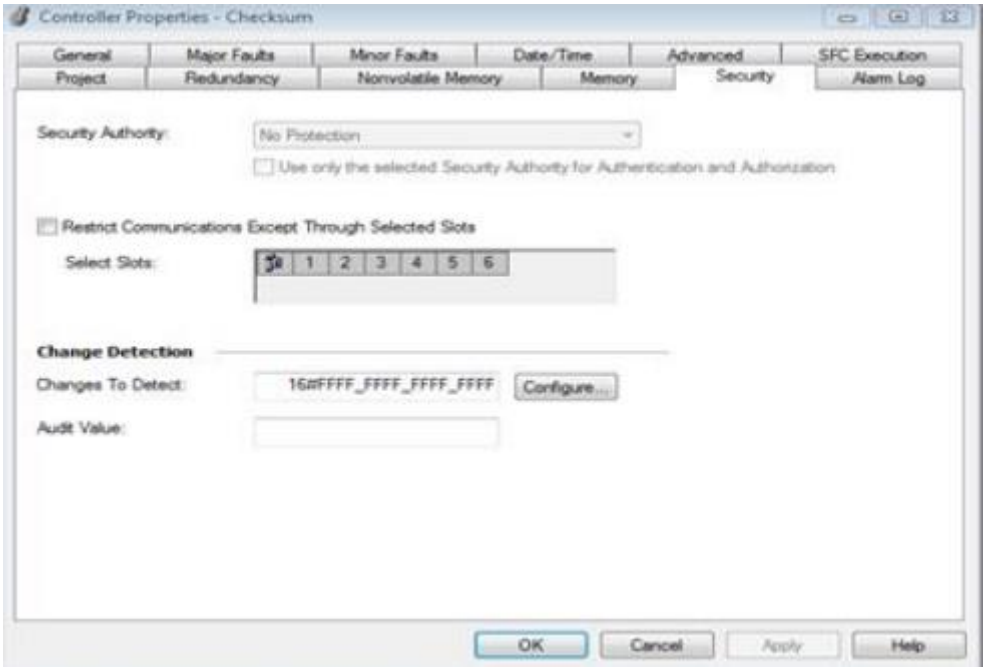
	Date	UTC Time	Referenz	Aktuell
1	11/21/2019	9:55:11	84 2A 76 DF 5B 31 F4 16	FF 2C EA 71 44 D7 81 04
2	11/21/2019	9:57:33	FF 2C EA 71 44 D7 81 04	FF 2C EA 71 44 D7 81 04
3	11/21/2019	9:58:17	FF 2C EA 71 44 D7 81 04	5B 7C 57 7E E2 3E EF C3
4	11/21/2019	9:58:36	FF 2C EA 71 44 D7 81 04	5B 7C 57 7E E2 3E EF C3
5	11/21/2019	9:58:44	5B 7C 57 7E E2 3E EF C3	5B 7C 57 7E E2 3E EF C3

罗克韦尔实现的例子:

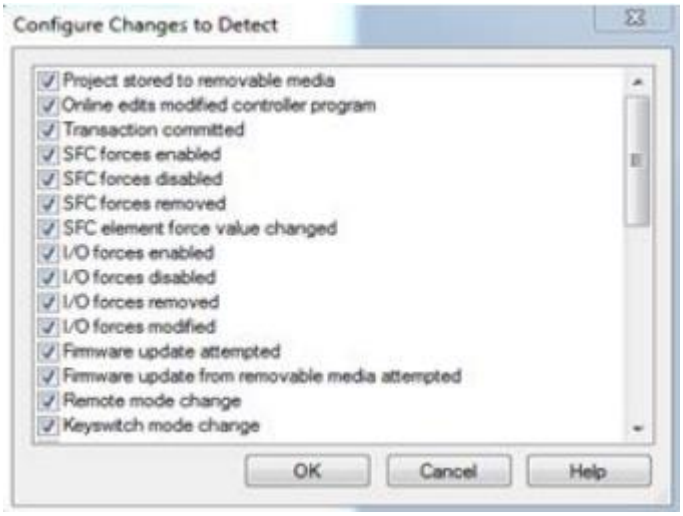
这是企业如何在其 ICS 环境中开发 PLC 程序变更检测能力的部分例子。这个例子只是针对罗克韦尔自动化控制 ControlLogix PLC，并不完整;然而，它说明了如何将 PLC 处理器状态获取到 PLC 内的寄存器中。一旦在 PLC 中注册，就可以使用它创建一个配置更改告警并且显示在 HMI 上，将原始状态信息发送到 HMI 进行趋势分析和监控，或将其发送给历史数据库进行长期捕获记录。

该实践提供了一个利用现有的工具的机会，可以获得关键网络资产变化时的态势感知，并在其环境中以最适合的方法来使用。

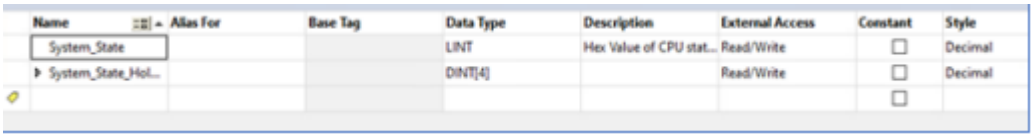
1. 在控制器属性对话框中，选择“Change to Detect”上的配置按钮



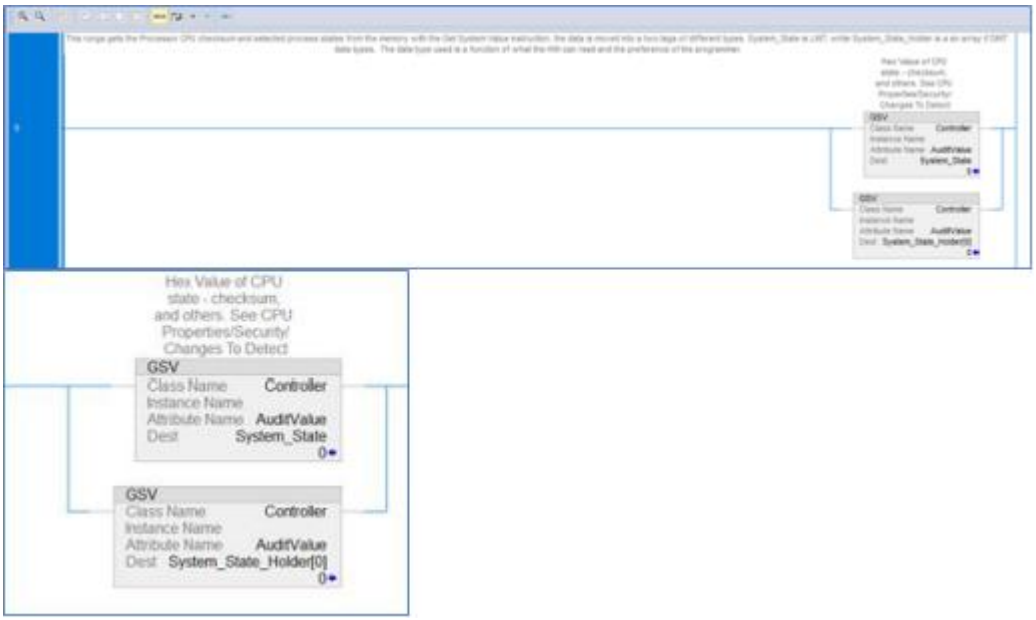
2. 在选择窗口中，选择要监控的所有项目



3. 创建一个 Tag 来接收处理器状态信息。这个标签可以是类型" LINT "，也可以是类型" DINT "



4. 使用获取系统值(GSV)指令从内存中获取处理器状态信息，并将其移动到可以在逻辑中使用或在 HMI 中读取的标签中.



优点与好处

维度	原因
安全性	知道 PLC 代码是否被篡改对于发现攻击和验证被攻击后 PLC 是否能够安全运行都是至关重要的。
可靠性	哈希或校验和是验证 PLC 是否(仍然)运行的一种手段，这需要在集成商/制造商提供支持功能的前提下完成。
运维	/

参考材料

标准/框架名称	涉及内容
MITRE ATT&CK for ICS	Tactic: TA002 - Execution, TA010 - Impair Process Control Technique: T0873 - Project File Infection, T0833 - Modify Control Logic
ISA 62443-3-3	SR3.4 : Software and information integrity
ISA 62443-4-2	CR3.4 : Software and information integrity

标准/框架名称	涉及内容
	EDR3.12 : Provisioning product supplier roots of trust
ISA 62443-4-1	SI-1 : Security implementation review SVV-1 Security requirements testing
MITRE CWE	CWE-345: Insufficient Verification of Data Authenticity • (child) CWE-353: Missing Support for Integrity Check • (child) CWE-354: Improper Validation of Integrity Check Value

6. 验证定时器和计数器

如果定时器和计数器的值被写入 PLC 程序，PLC 应该验证写入值的合理性，并且验证输入值小于零时计数器反向计数情况。

安全目的	目标群体
PLC 逻辑的完整性	**集成/维护服务提供商资产所有者**

指导

计时器和计数器在技术上可以被预设为任何值。因此，需要限制预设计时器或计数器的有效范围，以满足操作要求。

如果远程设备，例如 HMI，需要将计时器或计数器值写入到程序中:

- 不要让 HMI 直接写入计时器或计数器，而是通过一个验证逻辑后再写入到管脚
- 验证 PLC 中的预设数值和超时数值

在 PLC 中很容易直接进行计时器和计数器输入的验证(不需要任何能够进行深度数据包检查（DPI）的网络设备)，因为 PLC"知道"进程状态或上下文是什么。它可以验证获取的内容及何时获取命令或设定的值。

示例

在 PLC 启动过程中，计时器和计数器通常预置一个特定数值。

如果有一个计时器在 1.3 秒时触发警报，但是这个定时器被恶意地预设为 5 分钟，那么它可能就不会告警。

如果有一个计数器在达到 10,000 时会让进程停止，但从一开始就将其设置为 11,000，那么进程可能就不会停止。

优点与好处

维度	原因
安全性	**如果 I/O、计时器或预设值被直接写入 I/O，而不被 PLC 验证，则 PLC 验证层会被绕过，HMI(或其他网络设备)会被分配一个不合理的信任级别。**
可靠性	**当操作员意外预设不合理的定时器或计数器值时，PLC 也可以验证。**
运维	**记录并自动验证计时器和计数器的有效范围有助于更新逻辑。**

参考材料

标准/框架名称	涉及内容
MITRE ATT&CK for ICS	Tactic：TA010 - Impair Process Control Technique: T0836 - Modify Parameter
ISA 62443-3-3	SR3.5 : Input Validation
ISA 62443-4-2	CR3.5 : Input Validation
ISA 62443-4-1	SI-2 : Secure coding standards SVV-1 : Security requirements testing

7. 对成对的输入/输出进行验证和警报

如果有成对的信号，请确保两个信号没有同时被使用或者激活。当出现物理上不可行的输入/输出状态时，应向操作员发出告警。当切换输出可能损害执行器时，应考虑独立处理对应的信号或添加延迟计时器以避免出现事故。

安全目的	目标群体
PLC 变量的完整性/弹性	**产品供应商、集成/维护服务提供商**

指导

成对的输入或输出是物理上不能同时发生的值，它们是排斥的。尽管成对的信号不能同时使用或者激活，除非出现故障或恶意攻击行为，但 PLC 程序员通常不会阻止这种情况的发生。

验证是最容易且可以直接在 PLC 中执行，因为 PLC 知道处理流程或上下文状态。如果成对的信号的地址具有一定的顺序性(例如输入 1 和输入 2)，则更容易识别和跟踪。

成对的输入或输出可能导致问题的另一种情况是，它们不是同时使用或者激活的，而是以一种损坏驱动器的方式快速切换。

示例

成对信号的例子:

启动和停止

> 独立的启动和停止:配置启动和停止作为离散量输出，而不是有一个单独的输出，可以切换/关闭。涉及到 HMI 或者更高层级的界面组态时应该禁止同时触发同一个输出变量。对于攻击者来说，如果必须设置两种不同的输出，则快速切换开/关要复杂得多。
>
> 重启计时器:在停止运行后发增加一个重启计时器，以避免快速切换启动/停止造成的执行机构的破坏。

正向和反向

打开和关闭

切换成对信号行为可能造成的破坏:

如果 PLC / MCC 接受离散输入，这为攻击者提供了一个对执行器造成物理损伤的机会。切换输出而造成破坏的常见场景是 MCC，但此实践适用于切换输出可能造成破坏的所有场景。2007 年爱达荷国家实验室进行的极光发电机测试证明了快速切换输出可能造成的实际损害，在该测试中切换输出不同步导致了断路器的损坏。

优点与好处

维度	原因
安全性	**1. 如果 PLC 程序没有考虑两个成对的输入信号同时激活所造成的后果，这会导致其成为一个很好的攻击向量。** **2. 如果存在激活的两个成对输入信号，这是一种告警，说明存在操作错误、编程错误或发生了一些恶意攻击行为。** **3. 这避免了可能对执行器造成物理损坏的攻击场景。**

维度	原因
可靠性	1. 成对的输入信号可能表明传感器坏了或接线错误，或存在机械问题，如开关卡住。 2.快速切换启动和停止可能是错误的，所以这也可以防止误操作造成的损害。
运维	/

参考材料

标准/框架名称	涉及内容
MITRE ATT&CK for ICS	Tactic: TA010 - Impair Process Control Technique: T0836 - Modify Parameter, T0806 - Brute Force I/O
ISA 62443-3-3	SR3.5: Input Validation SR3.6: Deterministic Output
ISA 62443-4-2	CR3.5: Input Validation CR3.6: Deterministic Output
ISA 62443-4-1	SI-2: Secure coding standards SVV-1: Security requirements testing
MITRE CWE	CWE-754:Improper Check for Unusual or Exceptional Conditions

8. HMI 的输入变量应该同时在 HMI 和 PLC 层面进行验证

HMI 对 PLC 变量的访问可以(也应该)限制在 HMI 的有效操作值范围内，但应在 PLC 中添加进一步的验证检查，以防止接收超出范围的值，如果出现应及时发出告警。

安全目的	目标群体
PLC 变量的完整性	**产品供应商，集成/维护服务提供商**

指导

输入验证可以包括有效操作值的越界检查，以及针对流程的数据类型的有效值。

如果一个 PLC 变量接收到一个越界的值，要确保 PLC 有以下其中一项逻辑:

- 为该变量输入一个默认值，该值不会对流程产生负面影响，可以用作告警标志
- 将最后一个正确的值输入到该值中，并记录事件以便进一步分析。

示例

例子 1

用户需要在 HMI 上输入阀门压力的值。此操作的有效范围为 0-100，用户的输入从 HMI 上的用户输入函数传递到 PLC 中的 V1 变量。在这种情况下，

1. HMI 输入变量 V1 有一个限制范围 0-100。
2. PLC 有一个交叉验证逻辑。

如果 v1 < 0 或 v1 > 100，则设置 v1 = 0。

从而确保该变量被赋予一个不合理的数值时，可以提供一个在安全范围内的正常响应。

例子 2

用户需要输入一个变量的测量阈值，该变量应该始终在 INT2 数据范围内。用户输入从 HMI 传递到 PLC 中的 V2 变量，这是一个 16 位的数据寄存器。

1. HMI 对变量 V2 的输入有一个限制范围-32768 到 32767(降序)。
2. PLC 具有数据类型的交叉验证逻辑，用于监控溢出的变量(V3)，该变量在 PLC 的内存结构中位于 V2 之后。

IF V2 = -32768 OR V2 = 32767 AND V3 != 0 ,

SET V2 = 0 AND V3 = 0 AND DataTypeOverflowAlarm = TRUE.

例子 3

将 PID（比例、积分、微分控制器）的 PV（过程值）、SP（设定值）和 CV（控制变量）的标度设置为一致或使用原始单位，以消除导致控制问题的标度误差。不正确的比例可能会导致潜在的被滥用情况。

优点与好处

维度	原因
安全性	**1. 虽然 HMI 通常会提供某种输入的验证，但恶意的攻击者可以制作或重放修改过的数据包，将任意值发送给 PLC 中对外部开放的变量(例如，从 HMI 传递到 PLC 内的数值)。** **2. PLC 协议通常被作为“开放”协议并发布给公众，因此开发“开放”协议信息的恶意软件是很容易的。PLC 变量映射通常可以在攻击的侦察阶段通过流量分析来实现，从而为入侵者提供必要的信息来制造到攻击目标的恶意流量，从而使用未经授权的工具操纵控制过程。在将该数据传送至相关进程之前，交叉检查验证传递到 PLC 的数值，确保其在有效的数据范围，并通过在 PLC 扫描过程中检测到值越界时，可以强制设置安全范围来缓解内存中的无效数值。**
可靠性	/
运维	/

参考材料

标准/框架名称	涉及内容
MITRE ATT&CK for ICS	Tactic: TA010 - Impair Process Control Technique: T0836 - Modify Parameter
ISA 62443-3-3	SR3.5: Input Validation SR3.6: Deterministic Output
ISA 62443-4-2	CR3.5: Input Validation CR3.6: Deterministic Output
ISA 62443-4-1	SI-2: Secure coding standards SVV-1: Security requirements testing
MITRE CWE	CWE-1320: Improper Protection for Out of Bounds Signal Level Alerts

9. 间接验证

通过对数组的溢出或异常下标输入来验证间接指令，以捕捉“栅栏柱（**fence-post**）”错误。

安全目的	目标群体
PLC 变量的完整性	**产品供应商，集成/维护服务提供商**

指导

间接寄存器是在另一个寄存器中使用这个寄存器的值。使用间接验证有很多理由。

- 可变频率驱动器(VFD)，使用查找表触发不同频率的不同动作。
- 根据当前泵的运行时间来决定先运行哪个泵

PLC 通常没有一个“数组结束”标志，所以在软件中创建它是一个好选择，可以用来避免异常/计划外的 PLC 操作。

示例

指令列表(IL)编程，这种方法可以转换成几个功能块，甚至可能被其他应用程序所重用。

1. 创建数组的掩码（array mask）

检查数组是否为二进制大小。如果它不是二进制大小的，创建一个掩码使其达到下一个二进制大小。

例如，如果你需要 5 个寄存器(非二进制大小):

[21 31 41 51 61]

定义一个 8 个元素的数组:

[x x 21 31 41 51 61 x]

接下来，取该间接性的索引值——在本例中是第 3 个元素。注意:索引从 0 开始!

[21 31 41 51 61]

_____^

索引值:3

添加一个偏移量来避免“尾端中毒”情况。偏移量可以是 1 或更高，在这种情况下偏移量是 2:

[x x 21 31 41 51 61 x]

_____^

包括偏移量的索引:3 + 2 = 5

然后 AND 索引，包括偏移量和掩码等于数组大小。

在这个例子中，数组大小是 8，因此索引是 7，所以掩码是 0x07。掩码确保你能得到的最大索引是 7，例如:

6 和 0x07 会给出 6。

7 和 0x07 会给出 7

8 和 0x07 会给出 0。

9 和 0x07 会给出 1。

这确保在数组中总能寻址到一个值而不至于发生溢出的情况。

2. 插入 poisoned ends

poisoned ends 是可选的。您可以在不使用 poisoning 技术的情况下检测被操纵的转移因素，但是 poisoning 技术有助于捕获“栅栏柱（fence-post）”错误，会返回一个没有意义的值。

关键是在数组的索引 0 处，应该有一个无效的值——例如-1 或 65535。这是“poisoned ends”。同样，在数组的最后一个元素上也可以这样做。

所以，对于上面的数组，poisoning 的版本可能是这样的:

[-1 -1 21 31 41 51 61 -1]

3.记录不带掩码的间接地址值

然后记录不带 AND 掩码和偏移量的间接地址的值。在这个例子中，索引 3 将记录数值 51。

[21 31 41 51 61]

_____^

_____Index 3

4. 执行 AND 掩码并比较值(=间接验证)

将您记录的值与完成偏移量和 and 掩码后的值进行比较。

- 案例 A:正确的间接验证

  首先,确定 offset:

  Index + Offset = 3 + 2 = 5

  第二,确定 mask:

  5 AND 0x07 = 5

  第三,确定 indirection check:

  [-1 -1 21 31 41 51 61 -1]

  _____^

  包括偏移量的索引:5

  Value 为 51 等于记录的值，所以一切正常。

- 案例 B:间接操纵

如果你现在有一个间接控制，假设是 7，让我们看看会发生什么:

首先,确定 offset:

Index + Offset = 7 + 2 = 9

第二,确定 mask:

AND 0x07 = 1

第三,确定 indirection check:

[-1 -1 21 31 41 51 61 -1]

___^

包括偏移量的索引:1

Value = -1，此时并不等于记录的值，而且还表明您的 poisoned ends，因此您可以知道有人恶意操控了这个值。

5. 执行错误/程序员警报

如果此验证值与记录的值不同，那么应该触发软件质量位告警。

然后，检查间接值。如果它是一个 poisoned value，应该发出另一个软件质量位警报。这是一个 fence-post 错误。

优点与好处

维度	原因
安全性	**大多数 PLC 没有任何处理数组越界索引的功能。indirection 错误可能导致两种潜在的危险情况: 其一，如果一个间接操作导致从错误的寄存器中读取数据，那么程序将使用错误的值执行后续的逻辑过程。 其二，如果错误的间接信息导致写入错误的寄存器，程序将覆盖您想要保留的代码或值。在这两种情况下，间接错误可能很难发现，并可能产生严重影响。它们可能是人为误操作造成的，但也可能是被恶意插入的。**
可靠性	**识别编程中的非恶意人为误操作。**
运维	**/**

参考材料

标准/框架名称	涉及内容
MITRE ATT&CK for ICS	Tactic: TA010 - Impair Process Control Technique: T0836 - Modify Parameter
ISA 62443-3-3	SR3.5: Input Validation SR3.6: Deterministic Output
ISA 62443-4-2	CR3.5: Input Validation CR3.6: Deterministic Output
ISA 62443-4-1	SI-2: Secure coding standards SVV-1: Security requirements testing
MITRE CWE	CWE-129: Improper Validation of Array Index

10. 通过函数分配指定的寄存器块(读/写/验证)

为特定函数分配指定的寄存器块，以验证数据，避免缓冲区溢出和阻止未授权的外部写操作，以保护控制器数据。

安全目的	目标群体
PLC 变量的完整性	**产品供应商，集成/维护服务提供商**

指导

临时内存，也称为临时存储器，是一个很容易被利用的内存领域。例如，简单地写入一个越界的“Modbus”寄存器可能覆盖内存寄存器。

一般来说，寄存器存储器可以通过 PLC 网络被其他设备访问以进行读写操作。有些寄存器可以由 HMI 读取，有些可以由 SCADA 系统写入。拥有特定应用程序的特定寄存器阵列还可以更容易地(在控制器或外部防火墙中)配置来自另一个设备/HMI 的只读访问。

指定寄存器块有意义的函数示例如下:

- 读取
- 写入(从 HMI /控制器/其他外部设备)
- 验证写入
- 计算

确保向允许的寄存器进行外部写操作还有助于避免由于超出限制的逻辑执行或恶意入侵而导致的主体内存重置错误。这些指定的寄存器块可以被用作 I/O、计时器和计数器写入的缓冲区，通过验证缓冲区已被完全写入(不包含部分旧数据，部分新数据)和验证缓冲区中的所有数据。

背景:

主体内存和寄存器存储器的使用是不同的。主体内存用于存储当前正在执行的程序逻辑，而寄存器存储器被当前正在执行的逻辑用作临时存储器。虽然寄存器内存是一个临时内存，因为它正在被执行逻辑使用，它必然包含一些重要的变量，会影响主体逻辑。

示例

举例说明如果这种做法没有实现会发生什么:

(参考:G. P. H. Sandaruwan, P. S. Ranaweera, Vladimir A. Oleshchuk, PLC 安全和关键基础设施保护):

•西门子通常在标志位区域(从标志位 200.0 到标志位 255.7)使用 scratchpad memory 存储器。如果该区域内的位发生变化，根据该位或字节的重要性，PLC 有可能发生严重故障。

•假设攻击者可以访问 PLC 网络中的一台机器，并通过蠕虫感染该机器，该蠕虫能够将任意值写入寄存器内存。由于寄存器的内存值可以任意改变，所以它可以改变逻辑程序中的压力值或流量值。

•执行逻辑将基于更改设置一个新的值，这可能导致系统超过其安全范围，并可能导致严重故障。

实现这一实践的示例:

•在一个安全区(DCS 可以读取)的场景中，防火墙可以记录任何“写入”操作，因为这些寄存器在安全区中被定义为是只读的。

•在另一种情况下，可能会有一些写使能的寄存器，而其他的是只读的，但所有的只读寄存器在一个单一的阵列使它们更容易配置在控制器(或防火墙)中。

优点与好处

维度	原因
安全性	**1.容易通过函数保护控制器数据(读/写/验证)。 2.使拥有协议识别和解析功能的防火墙更容易完成它们的工作:规则变得更简单，因为 HMI 可以访问哪些寄存器块非常清楚。更容易管理防火墙中的(简单的)规则。 3.对内部临时内存进行未经授权的更改是一个很容易利用的漏洞(旁路逻辑攻击)。 4.当正确验证 PLC 程序的输入和输出时，任何更改(由恶意的入侵者或误操作)都可以很容易地被捕获，而不是长时间停留在逻辑序列中并在稍后执行时才会抛出错误/导致问题。**
可靠性	**1.使读写速度更快，因为减少了实现中交互的次数。 2.如果临时内存不受保护，即使经过授权的更改和编程错误也可能导致故障。 3.如果在处理之前没有检查数据的有效性，过长的消息报文在网络和通信时可能会导致意外错误。**
运维	**导致写入临时内存的编程错误会使排查定位变得困难，因此可以通过为写入分配特定的寄存器来避免这个问题。**

参考材料

标准/框架名称	涉及内容
MITRE ATT&CK for ICS	Technique: T0835 - Manipulate I/O image , T0836 - Modify Parameter
ISA 62443-3-3	SR3.4 : Software and information integrity SR3.5 : Input Validation SR3.6 : Deterministic Output
ISA 62443-4-1	SD-4: Secure design best practices SI-1: Security implementation review SI-2 : Secure coding standards SVV-1 : Security requirements testing
ISA 62443-4-2	CR3.4 : Software and information integrity CR3.5 : Input Validation CR3.6 : Deterministic Output
MITRE CWE	CWE-787: Out-of-bounds Write CWE-653: Insufficient Compartmentalization

11. 合理性检查工具

引用一种检查流程，利用交叉检查来验证输入数据或者逻辑的合理性。

安全目的	目标群体
I/O 值的完整性	**产品供应商，集成/维护服务提供商**

指导

有很多方法利用物理合理性来验证测量结果:

a)比较累计测量值和和时间无关的测量值

合理性检验可以通过在一段时间累计的值，并与独立于时间的测量值进行比较来完成。

b)比较不同的测量源

此外，用不同的方法衡量同一现象可以很好地检验其合理性。

不同的测量源不一定是不同的物理传感器，也可以使用替代的通信通道来完成(见示例)。

示例

a)比较累计测量值和独立于时间测量值

- 计量泵和罐体液位表:容积变化应等于综合流量。
- 锅炉内燃烧器:加热量应等于温度提升。

b)比较不同的测量源

- 利用飞机上的空气速度、水平线、垂直速度、高度来测量飞机上升/下降的现象。
- 比较独立数据记录器的工艺参数值(连接到 4-20mA 回路或继电器通断，并通过独立的通信通道传输)到 SCADA 系统数据(通过 PLC 和 HMI 的“正常”方式)，并对偏差和显著偏离规定的值发出警报。

优点与好处

维度	原因
安全性	**便于监控操作值(假设不是一次操作所有传感器)。**
可靠性	**防止接受或识别(为随后的操作)损坏/错误的测量值作为输入使用。**
运维	**更快地排除故障发生的物理层原因。**

参考材料

标准/框架名称	涉及内容
MITRE ATT&CK for ICS	Tactic: TA010 - Impair Process Control Technique: T0806 - Brute Force I/O
ISA 62443-3-3	SR3.5: Input Validation SR3.6: Deterministic Output
ISA 62443-4-2	CR3.5: Input Validation CR3.6: Deterministic Output
MITRE CWE	CWE-754: Improper Check for Unusual or Exceptional Conditions

12. 基于物理层面的合理性来验证输入

确保操作员在输入过程中只能输入实际有意义的数据。当出现偏差时或者无效输入时，应及时发出告警。

安全目的	目标群体
I/O 值的完整性	**产品供应商，集成/维护服务提供商**

指导

a)监测预期的物理过程持续时间

如果操作从一个极端到另一个极端所花费的时间比预期的要长，那就值得发出警报。同样的，如果这个时间太短，也需要发出告警。

一个简单的解决方案是步长超时 step-timeout 警报。这对于顺序/步骤控制的任务很有用。

例如，步骤“将物体从 A 移动到 B”从步骤开始到满足转换条件(传感器:物体到达 B)需要 5 秒。

如果条件满足的时间过早或过晚，步长超时将触发告警。

b)监控预期的物理过程的重复活动

物理过程的合理性检查意味着对违背物理现象的活动发出警报:如果预期有一个规律的，重复的事件循环(例如，批次，每天的模式)，那么当在预期变化(离散或模拟值)时，如果保持静态太长时间，则会发出警报。

示例

a)监测预期的物理过程持续时间

- 大坝的闸门从完全关闭到完全打开需要一定的时间
- 在废水处理系统中，储水井需要一定的时间来填满

b)监控预期的物理过程重复活动

- 制造过程或管道配料应在控制范围或操作模式之间定期循环。
- 城市污水处理厂的活动/进水流量模式通常有一个日循环。

c)限制操作员的设置，从而满足实际/物理上的可能情况。

例如，Oldsmar Florida 案例允许操作员输入:

a)比通常需要的值多数千倍;

b)这在物理过程上是不可能的。尽可能在 PLC 代码中配置操作限制，而不在 HMI 层面进行验证或者限制，因为还需要考虑带旁路攻击（不通过 HMI 也有可能使目标值过大或者过小）。

优点与好处

维度	原因
安全性	偏差可以表明执行器已经处于失控状态，或者有人试图伪造 I/O，例如通重放攻击。 非活动状态告警有助于监视冻结或强制的常数值。
可靠性	如果损坏设备源于电气或机械故障，该实践会提前发出告警。 非活动告警有助于标记测量或系统控制环路，这些环路可能由于物理设备故障或逻辑控制算法问题或操作员的错误输入而失效（因此是静态的）。
运维	/

参考材料

标准/框架名称	涉及内容
MITRE ATT&CK for ICS	Tactic: TA010 - Impair Process Control Technique: T0806 - Brute Force I/O
ISA 62443-3-3	SR3.5: Input Validation SR3.6: Deterministic Output
ISA 62443-4-2	CR3.5: Input Validation CR3.6: Deterministic Output
MITRE CWE	CWE-754: Improper Check for Unusual or Exceptional Conditions

13. 禁用不需要/不使用的通信端口和协议

PLC 控制器和网络接口模块一般支持默认开启的多种通信协议。应该禁用设备中不需要的端口和协议。

安全目的	目标群体
加固	**集成/维护服务提供商**

指导

通常默认启用的协议有:HTTP、HTTPS、SNMP、Telnet、FTP、MODBUS、PROFIBUS、以太网/IP、ICMP 等。

最佳实践是开发一个数据流程图，描述 PLC 和系统中其他组件之间所需的通信关系。

数据流程图应该显示 PLC 上的物理端口以及它们所连接的逻辑网络。对于每个物理端口，应该确定所需的网络协议列表，并禁用所有其他协议。

示例

例如，许多 PLC 都包含一个用于维护和故障排查的 web 服务器。如果不使用这个特性，应在条件允许的情况下禁用它，因为这可能是一个攻击向量。

优点与好处

维度	原因
安全性	**每个启用的端口和协议都会增加 PLC 的攻击面。确保攻击者不能使用它们进行未经授权的通信的最简单的方法是完全禁用它们。**
可靠性	**如果 PLC 不能通过某个端口或协议进行通信，这也会减少潜在的(畸形的)通信报文，无论是否是恶意的，这也会减少由于意外的/畸形的通信报文而导致 PLC 崩溃的机会。**
运维	**禁用未使用的端口和协议也便于维护，因为它降低了 PLC 的整体复杂性。不存在的内容不需要管理或更新。**

参考材料

标准/框架名称	涉及内容
MITRE ATT&CK for ICS	Tactic: TA005 - Discovery Technique: T0808 - Control Device Identification , T0841 - Network Service Scanning, T0854 - Serial Connection Enumeration
ISA 62443-3-3	SR 7.6: Network and security configuration settings SR 7.7: Least functionality
ISA 62443-4-2	EDR2.13 : Use of physical diagnostic and test interfaces

14. 限制第三方数据接口

限制第三方接口的连接类型和可用数据。应该对连接（和/或）数据接口进行适当定义，并将限制仅为所需数据进行读/写。

安全目的	目标群体
加固	**集成/维护服务提供商**

指导

在某些情况下，由于长电缆传输或大量数据报文的交换，接口数据交互模式比直连数据交换更好。

在设计和实现第三方数据交换接口时，应考虑并遵循以下指导原则:

- 使用专用的通信模块，可以直接与第三方 PLC 或数据交换设备相连，也可以使用与各方核心网络物理隔离的专用网络设备。
- 所连接设备的 MAC 地址通常在任何支持 ICS 以太网的设备的系统变量中可用，这使得通过多因素方法(IP 地址+ MAC 制造商代码=可信设备)验证设备标识成为可能。这种做法当然不是万无一失，因为 MAC 和 IP 地址可以被欺骗，但它提高了可信 ICS 系统和设备之间通信的门槛。
- 当为第三方接口选择协议时，选择一个协议，使第三方所有者的系统写入数据的可能性最小化。
- 选择一种连接方式和连接端口，以防止第三方能够配置所有者的 PLC 或数据交换设备。
- 第三方不应能够读取或写入尚未明确定义和可用的任何数据。
- 使用看门狗定时器监控通信过程，使通信命令不会发送到故障模式下的 PLC。
- 串行连接:使用一个专用的通信模块为每个第三方接口与一个限制的数据阵列。确保连接的所有者一方是发起者，第三方是响应者。
- 以太网/IP:一些 PLC 允许通信模块发挥防火墙的作用，并可以执行深度包检查(DPI)，或限制通信模块接口，以限制数据交换到一个预定义的子集中。如果这些特性可用，且使用了以太网/IP 协议，请确保已启用并配置这些功能。
- 当操作或合同需求阻止所有者完成前面的项目时，考虑使用一个单独的“数据集中器”(又名代理/DMZ) PLC，以缓冲数据并保护所有者免受第三方不必要的写入/编程。确保该 PLC 的背板不从第三方网络穿过。

示例

- 管道或租赁自动托管转移(LACT)单元，用于在上游生产或管道公司和中游管道公司之间传输和计量碳氢化合物或水，具有网络或串行接口连接，在公司之间共享计量、状态和许可信息。
- 地区饮用水供应商(进口商)共享道岔水流量被交付给当地市政当局的水厂。

优点与好处

维度	原因
安全性	**1. 限制暴露于第三方网络和设备。** **2. 认证外部设备，防止欺骗。**
可靠性	**限制有意或无意的修改或从第三方地点或设备的访问。**
运维	**/**

参考材料

标准/框架名称	涉及内容
MITRE ATT&CK ICS	Tactic: TA010 - Impair Process Control Technique: T0836 - Modify Parameter
ISA 62443-3-3	SR 7.6: Network and security configuration settings SR 7.7: Least functionality
ISA 62443-4-2	CR 7.6: Network and security configuration settings CR 7.7: Least functionality
ISA 62443-4-1	SD-4: Secure design best practices SI-1: Security implementation review SVV-1: Security requirements testing

15. 定义 PLC 重新启动时的安全状态

在 **PLC** 重新启动的情况下，定义一个重启后的安全状态**(**例如，通电，断电，保持以前的状态**)**。

安全目的	目标群体
抗逆力 （Resilience）	产品供应商，集成/维护服务提供商

指导

如果有什么指令可以让 PLC 在工作过程中重新启动，我们应该期望程序能够顺利启动，并且对工作过程的干扰最小。确保整个过程是安全重启的。

如果将 PLC 配置为安全重新启动状态是不现实的，请确保它会发出告警并且不会发出任何新的指令。此外，在这种情况下，确保标准操作程序(SOP)有非常明确的指示（设置手动控制），以便 PLC 将正常启动某程序。

同时，记录所有的启动、关闭、稳态控制等系统的重启程序。

示例

/

优点与好处

维度	原因
安全性	**消除潜在的意外行为:** **PLC 最基本的攻击方式是迫使它崩溃和/或重新启动。对于许多 PLC 来说，这并不难做到，因为许多 PLC 不能很好地应对意外的输入或太多的流量报文。当它运行时，有一些控制器的诊断信息，但是它如何处理一个正在运行的进程的重新启动状态通常是不清楚的。这可能不常见，但如果考虑到攻击者的恶意行为，这是一个很基础的攻击方式。**
可靠性	**避免意外延误:** **如果在 PLC 上电之后,状态机初始化状态和一些条件无法正常启动某过程控制,系统操作员不能更正系统,需要技术员访问 PLC 内部程序强制去更改流程状态才能够开始操作。这可能会导致延迟和生产损失。**
运维	**/**

参考材料

标准/框架名称	涉及内容
MITRE ATT&CK ICS	Tactic: TA009 - Inhibit Response Function Technique: T0816 - Device Restart/Shutdown
ISA 62443-3-3	SR3.6: Deterministic Output
ISA 62443-4-2	CR3.6: Deterministic Output
ISA 62443-4-1	SVV-1: Security requirements testing

16. 收集 PLC 的运行周期并在人机界面上进行趋势分析

每 2-3 秒汇总 PLC 的执行周期时间，并上传给人机界面以图形化的方式呈现。

安全目的	目标群体
监控	集成/维护服务提供商

指导

周期时间通常是 PLC 中的系统变量，可用于汇集 PLC 代码执行过程。应该利用这个功能来计算平均、峰值和最小周期时间。HMI 应将这些数值趋势化、可视化，并在有重大变化时发出警报。

循环时间是计算 PLC 每次逻辑迭代执行所需的时间。每次的迭代执行是梯形图(LD)、功能模块图(FBD)、指令列表(IL)和结构化文本(ST)的组合或者任意一种。这些逻辑组件可以与顺序功能图(SFC)连接在一起。

在 PLC 上的循环时间应该是恒定的，除非有变化，例如：

- 网络环境
- PLC 逻辑程序
- 控制过程

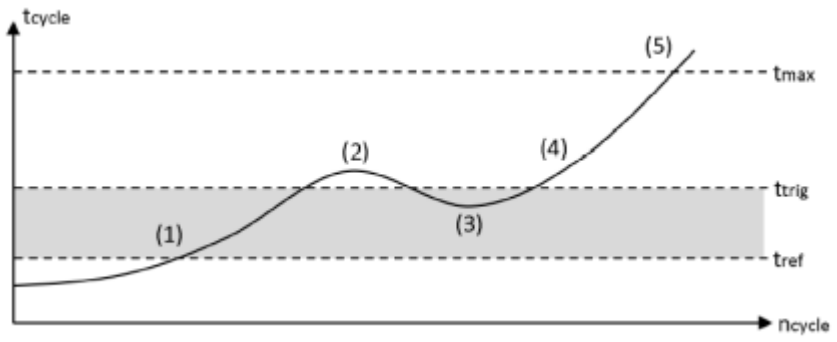
因此，异常的周期时间变化可以作为 PLC 逻辑变化的指示标志，从而为完整性检查提供有价值、有意义信息。

图表可视化提供了一种直观的方式来引起人们对异常现象的关注，而仅仅使用绝对值是很难注意到异常变化现象的。

示例

许多 PLC 在硬件级别有一个“最大周期时间”监控功能。如果周期时间超过最大值，硬件将 CPU 设置为 STOP 状态(5)。

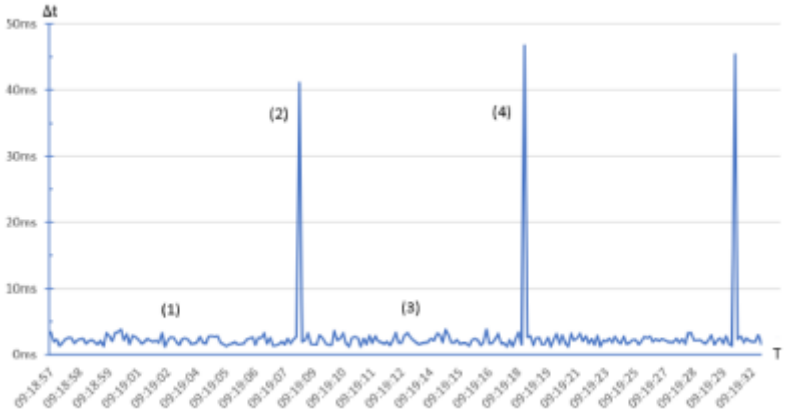
当然，攻击者意识到这一点，并将尽可能保持可能的攻击代码为最精简的状态，以尽量减少对整个周期时间的影响。在一个附加的软件周期时间监控程序中，参考周期时间 tref 被定义为基本周期时间。由于小波动是自然现象，因此需要定义一个可接受的阈值(1,3)。如果超过阈值(2,4)，则触发周期监控。



任何偏离参考时间的情况都可以存储在一个日志文件中，如下所示:

SeqNo	Date	UTC Time	Abweichung
1	2019-11-22	09:05:50.021	40,821ms
2	2019-11-22	09:06:00.069	44,391ms
3	2019-11-22	09:06:10.120	44,994ms
4	2019-11-22	09:06:20.166	40,561ms
5	2019-11-22	09:06:30.211	40,725ms

如果周期时间在 HMI 上做了可视化的呈现，那么就可以一眼看到 CPU 运行的确实。下面的示例图显示了一个定期执行恶意代码的 PLC 程序。(1,3)在正常运行时显示可接受的周期时间波动(“噪声”)，攻击代码在(2,4)上执行，增加了循环周期时间。



优点与好处

维度	原因
安全性	对 PLC 的攻击包括改变其逻辑、激活一个新程序、测试新代码、加载一个新的过程配方、插入辅助逻辑来发送消息或激活某些功能。对于大多数 PLC 来说，传统的密码完整性检查是不可行的。但是，如果发生上述任何逻辑变化，最好是需要发出警告的。由于在正常情况下，循环时间是相当恒定的，因此循环时间的变化是一个很好的标志，表明上述逻辑组件中的逻辑发生了变化。
可靠性	同安全性，但不是出于恶意原因
运维	/

参考材料

标准/框架名称	涉及内容
MITRE ATT&CK ICS	Tactic: TA002 - Execution Technique: T0873 - Project File Infection
ISA 62443-3-3	SR3.4: Software and information integrity
ISA 62443-4-2	EDR3.2: Protection from malicious code

标准/框架名称	涉及内容
MITRE CWE	CWE-754:Improper Check for Unusual or Exceptional Conditions

17. 记录 PLC 的正常运行时间并在人机界面上进行趋势分析

记录 **PLC** 的正常运行时间，以知道它重新启动的时间。在 **HMI** 上追踪和记录正常运行时间，用以进行诊断。

安全目的	目标群体
监控	**集成/维护服务提供商**

指导

跟踪 PLC 的运行时间

- (如果正常运行时间是 PLC 中的一个系统变量)
- 在 PLC 本身如果它有 MIB-2 /SNMP 实现
- 外部通过 SNMP 等方式

如果 PLC 上有带 MIB-2 的 SNMP，这是很常见的情况，则 uptime 的 OID 为“sysUpTimeInstance(0)”的前缀为 1.3.6.1.2.1.1.3。正常运行时间复位是 PLC 重新启动的重要指标。确保 HMI 对任何类型的 PLC 重启发出警报。

与错误代码相关的正常运行时间是很好的诊断方法。

示例

优点与好处

维度	原因
安全性	**PLC 最基本的攻击方式是使它崩溃和/或重新启动。对于许多 PLC 来说，这并不难做到，因为许多 PLC 不能很好地应对意外的输入或太多的流量报文。因此，意外重启可能是 PLC 遇到异常操作的指示标志。**
可靠性	**PLC 重启还有助于诊断故障，并监控哪些 PLC 在什么时间正在工作。**
运维	**/**

参考材料

标准/框架名称	涉及内容
MITRE ATT&CK ICS	Tactic: TA009 - Inhibit Response Function Technique: T0816 - Device Restart/Shutdown
ISA 62443-3-3	SR7.6: Network and security configuration settings
ISA 62443-4-2	CR7.6: Network and security configuration settings

标准/框架名称	涉及内容
MITRE CWE	CWE-778: Insufficient Logging

18. 在 HMI 上记录 PLC 硬停止并且进行趋势分析

在 HMI 上存储由故障或关机造成的 PLC 停止事件，PLC 重启前，由 HMI 发出告警。同时利用时间同步来获取更准确的数据。

安全目的	目标群体
PLC 逻辑的完整性监控	**集成/维护服务提供商**

指导

故障事件表明 PLC 关闭的原因，以便在重新启动前解决问题。

一些 PLC 可能有来自上次 PLC 故障或不正常关机的错误代码。记录这些错误，然后清除它们。如果存在这些特性和基础设施，那么最好将这些错误作为信息数据报告给 HMI，或者报告给 syslog 服务器。

大多数 PLC 还具有某种类型的首周期扫描功能，可以生成事件。这是几乎所有的 PLC 设备都有某种形式的行为。它基本上是一个或多个标志，或在 PLC“唤醒”后的第一次扫描上执行的指定例程序。应该记录并跟踪第一次扫描过程。

示例

/

优点与好处

维度	原因
安全性	**日志可以在发生事件时进行故障排除。在 PLC 开始运行之前，特别是在遇到问题之后，确保它是可信的是很重要的。**
可靠性	**如果事件不是恶意引起的，日志也是很好的排查故障的第一手资料。**
运维	**/**

参考材料

标准/框架名称	涉及内容
MITRE ATT&CK ICS	Tactic: TA009 - Inhibit Response Function Technique: T0816 - Device Restart/Shutdown 1
ISA 62443-3-3	SR7.6: Network and security configuration settings
ISA 62443-4-2	CR7.6: Network and security configuration settings
MITRE CWE	CWE-778: Insufficient Logging

19. 在 HMI 上监控 PLC 内存的使用情况并进行趋势分析

测量部署在生产环境中的每个控制器的内存使用情况并且建立一个基线，同时在 **HMI** 上对其进行趋势分析。

安全目的	目标群体
PLC 逻辑的完整性监控	**集成/维护服务提供商，资产所有者**

指导

由于逻辑程序中代码行数的增加也会导致运行时内存消耗的增加，因此建议 PLC 程序员跟踪任何偏离基线的情况，并为这个事件专门设置一个报警专属类。

示例

在 Rockwell Allen Bradley PLC 中，可以在控制器上建立基线，使用 RSLogix 5000 任务监控工具可以跟踪内存使用情况。不仅需要跟踪主体内存，而且 I/O 内存和 Ladder/Tag 内存也可以使用趋势进行跟踪。

优点与好处

维度	原因
安全性	**内存使用量的增加可以作为 PLC 运行更改代码的指示标志。**
可靠性	**跟踪运行程序的内存使用情况有助于避免总内存消耗和 PLC 控制器的故障状态。**
运维	**跟踪内存使用可以用于调优和为被监视的控制器找到最佳扫描时间，还可以用于故障排除和故障状态相关问题的调查。**

参考材料

标准/框架名称	涉及内容
MITRE ATT&CK ICS	Tactic: TA002 - Execution Technique: T0873 - Project File Infection
ISA 62443-3-3	SR3.4: Software and information integrity
ISA 62443-4-2	EDR3.2: Protection from malicious code

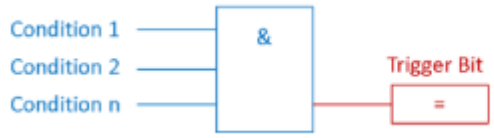
20. 捕捉关键警报的假阴性和假阳性

识别关键告警并为这些告警设置门限 trap。通过门限 trap 来监控触发条件和有变化的告警状态。

安全目的	目标群体
PLC 逻辑的完整性监控	**集成/维护服务提供商**

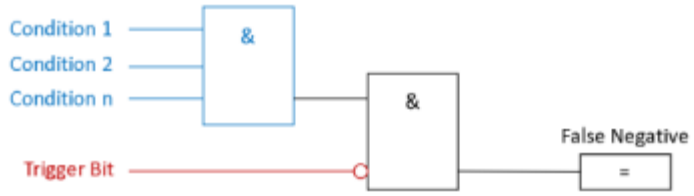
指导

在大多数情况下，警报状态是布尔值(True, False)，并由特定条件触发，如下所示。例如，如果条件 1“压力开关 1”，条件 2“压力传感器值超过临界阈值”，通过 n.为 TRUE，警报“超压”的触发位变为 TRUE。



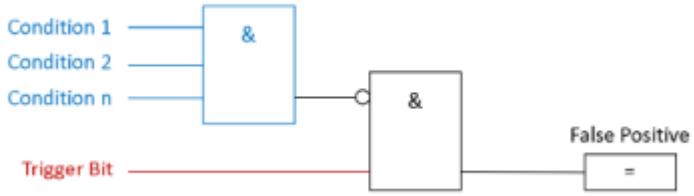
为了伪装攻击，攻击者可以抑制警报触发位并造成假阴性。

假阴性的陷阱监视触发位和否定触发位本身的条件。通过这个简单的设置，可以检测到假阴性。如图所示:



在其他情况下，攻击者可能故意造成假阳性，以转移操作员的注意力。

与假阴性陷阱相同，通过监控报警触发位，如果触发条件满足，也可以检测到假阳性。如果不满足条件，但触发位激活，则检测为假阳性。如下图所示:



示例

例 1:

西门子在其 Siemens S7-1200/1500 产品中提供了一个具有广泛功能的 web 服务器，例如显示 PLC 状态、周期时间或范围的记录。它还可以查看和修改数据表和变量。web 服务器的访问权

限可以在“PLC -硬件设置”中修改。在错误配置访问权限的情况下，对手可以获得对 PLC 变量和数据锁的访问。为了创建一个假阳性，对手选择一个警报触发位并改变状态。

例 2:

在 Triton/Trisys/HatMan 攻击中，恶意攻击代码抑制了警报状态。

例 3:

总线注入攻击可能向高层级的 SCADA 客户端发送假阳性警报。

优点与好处

维度	原因
安全性	**减轻攻击(例如，恶意攻击代码、总线注入、篡改不安全的 web 服务器上可访问的 PLC 状态表)引起的严重警报消息的假阴性或假阳性。**
可靠性	**/**
运维	**/**

参考材料

标准/框架名称	涉及内容
MITRE ATT&CK ICS	Tactic : TA009 - Inhibit Response Function Technique: T0878 - Alarm Suppression
ISA 62443-3-3	SR 3.5 : Input Validation
ISA 62443-4-2	CR 3.5 : Input Validation
ISA 62443-4-1	SI-1 : Security implementation review
MITRE CWE	CWE-754: Improper Check for Unusual or Exceptional Conditions

关于本 PLC 安全编码项目

多年来，可编程逻辑控制器(PLC)的设计是不安全的。经过几年的参考和借鉴 IT 领域的最佳实践，陆续出现了安全协议、加密通信、网络隔离等安全措施。然而，到目前为止，还没有人关注如何使用 PLC(或 SCADA/DCS)中的特定功能来实现安全性，或者如何在考虑安全性的情况下进行 PLC 的编程。这个项目——受到现有 IT 安全编码实践的启发——填补了这一空白。

谁应该阅读和应用本 PLC 安全编码实践?

这些实践是为工程师编写的。这个项目的目的是为正在创建编程逻辑(梯形逻辑，功能图等)的工程师提供指导，以帮助改善工业控制系统的安全态势。这些实践利用了 PLC/DCS 中本身可用的功能。几乎不需要额外的软件工具或硬件来实现这些实践。

它们都能适应正常的 PLC 编程和操作流程。要实现这些实践，不仅需要安全专业知识，还需要对目标 PLC、逻辑语言和底层控制过程有深入的了解。

PLC 编码和该实践的应用范围是什么?

如要应用该 TOP20 PLC 安全编码实践，需要直接对 PLC 进行更改。本文档中只是大量可能的 PLC 安全编码实践的 TOP 20 的一部分。另外还有一些与总体架构、人机界面或文档相关的实践草案。这些不适合 PLC 安全编码的范围，但未来可应用于构建安全的 PLC 运行环境。

应用 PLC 安全编码实践的好处是什么?

使用这些实践显然有安全方面的好处——主要是减少攻击面或在发生安全事故时能够更快地排查故障。然而，许多实践除了安全性之外还有其他好处。其中的一些还可能使 PLC 代码更加可靠，更容易调试和维护，更容易通信，可能也更加精简。此外，PLC 安全编码实践不仅可以在发生安全攻击时保护用户，而且可以使 PLC 代码更加健壮，以承受意外的配置错误或人为误操作。

该项目的参与者是谁?

这一切都始于 Jake Brodsky 的 S4x20 演讲"PLC 的安全编码实践"。

演讲后，Dale Peterson 发起了 TOP20 项目。Jake Brodsky 和 Sarah Fluchs 花了几个小时打电话去讨论这个事情，把 Jake 提出的 PLC 安全编码方法写在纸上。随后，Dale、Jake 和 Sarah 在 top20.isa.org 上建立了一个平台，由 ISA GCA 支持，从 ICS 安全和工程师社区组织和收集额外的内容。

讨论和迭代安全实践的文档，并策划一个合适的 TOP 20 清单大约用了一年的时间;Vivek Ponnada 的帮助大大加速了这一过程，他除了贡献和审查内容，还组织了定期的电话会议，直到所有实践的建议都得到解决，Mohamed Abdelmoez Sakesli 尽最大努力完成了所有标准参考文献的内容，MITRE CWE 团队在最后时刻提供了 CWE 参考文献，Sarah，Jake, Dale, John Cusimano, Dirk Rotermund, Josh Ruff, Thomas Rabenstein, Gus Serino, Walter Speth, Agustin Valencia Gil-Ortega, Marcel Rick-Cen, and Al Ratheesh R,，他们会在定期的电话讨论中逐步完善本文档的相关信息。

支持者的名单

PLC 安全编码项目现在以及将来都需要整个社区的努力，如果没有无数贡献者慷慨地分享他们的时间和 PLC 方面的安全知识，这个项目是不可能完成的。共有 943 名用户在平台上注册参与

相关的讨论。以下是按字母顺序列出的所有明确同意被提及的人。感谢所有耗费时间和精力支持这个项目的人!

Aagam Shah
Adam Paturej
Agustin Valencia Gil-Ortega
Aitor García Almiñana
Alec Summers
Al Ratheesh. R
Andreas Falk
Anton Shipulin
Arkaitz Gamino
Carlos Olave
Chris van den Hooven
Chris Sistrunk
Christos Alexopoulos
Cris DeWitt
Dale Peterson
Dene Yandle
Dennis Verschoor
Dirk Rotermund
Edorta Echave García
Gananand Kini
George Alex Holburn
Gus Serino
Josie Houghton
Jozef Sulwinski
Juan Pablo Angel Espejo
Khalid Ansari
Marc Weber
Marcel Rick-Cen
Martin Huddleston
Massimiliano Zonta
Matthew Loong
Matthias Müller
Michael Thompson
Michal Stepien
Miguel Angel Frias
Mohamed Abdelmoez Sakesli
Moon Eluvangal Chandran
Nahuel Iglesias
Nalini Kanth
Narasimha S. Himakuntala
Omar Morando
Oscar J. Delgado-Melo
Päivi Brunou
Peter Donnelly

Hakija Agic	Peter Jackson
Hector Medrano	Ravindra Deshakulakarni
Heiko Rudolph	Rick Booij
Isiah Jones	Robert Albach
Jacob Brodsky	Rushi Purohit
Javier Perez Quezada	Sarah Fluchs
J-D Bamford	Sergei Biberdorf
Joe Weiss	Stephan Beirer
John Cusimano	Steve Christey Coley
John Hoyt	Thomas Rabenstein
John Powell	Tim Gale
John Kingsley	Vivek Ponnada
Joseph J. Januszewski	Vytautas Butrimas
Josh Ruff	Walter Speth

特别感谢以下组织慷慨地提供他们的设备和资源来协助项目组，例如提供域名、服务器以及网页设计和图形设计等。

License